SUBCOMMITTEE ON CYBERSECURITY AND INFRASTRUCTURE PROTECTION
ONE HUNDRED AND NINETEENTH CONGRESS

———————————————-

22 JULY 2025

———————————

Kim Zetter

Chairman Garbarino, Ranking Member Swalwell and distinguished members of the Subcommittee, thank you for giving me an opportunity to testify before you today on the subject of Stuxnet and threats to critical infrastructure. My name is Kim Zetter, and I'm a journalist, author and adjunct professor at Georgetown University. I've been writing about cybersecurity and national security for two decades as a staff writer for *Wired* magazine and as a freelancer for the *New York Times*, Politico, the *Washington Post* and others. I wrote what is considered to be the seminal work on Stuxnet — *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapo*n. Recently, I have also begun to teach graduate students about nation-state cyber operations — the threat actors behind them, the technical capabilities they use to pull off these often very sophisticated operations, and the vulnerabilities in critical infrastructure and other systems that make the operations possible. Many of my students currently hold positions in the federal government or military, and others plan to work in government when they complete their degrees. My goal is to provide them with a solid foundation of knowledge — both technical and contextual — that will serve them in the policy and decision making positions they currently hold or may hold one day.

Today, I've been asked to talk about the digital weapon known as Stuxnet, which was designed to sabotage Iran's nuclear program and was discovered in 2010, three years after it was unleashed. I've been asked specifically to describe how Stuxnet conducted its sabotage, the impact it had on Iran's nuclear systems, the implications for other critical infrastructure here in the US and whether these

systems are any more secure against similar attacks today than they were at the time Stuxnet was unleashed.

Fifteen years ago this month, Stuxnet was discovered on systems in Iran but its impact has not diminished and is still felt in the security community today.

Stuxnet was discovered after it spread out of control and far beyond the facility at which it was aimed. Although Stuxnet spread to millions of machines — the exact number is unknown — it only unleashed its destructive payload on the specific systems it was designed to target: systems at Iran's underground uranium enrichment plant at Natanz. It didn't sabotage other systems because Stuxnet was a highly sophisticated, carefully crafted, precision weapon that was designed to avoid collateral damage. Attacks against critical infrastructure, however, don't need to be precision-targeted or sophisticated to cause disruption or damage. They just need systems that are vulnerable. This is worth noting given the recent warnings from government about the potential for Iran to launch retaliatory cyberattacks against US critical infrastructure, following the recent US bombing of Iranian targets, including the Natanz facility that Stuxnet hit more than a decade ago. Iran doesn't have the skills to pull off a Stuxnet-like attack, but it doesn't need them to cause disruption and damage to US critical infrastructure.

Although a lot has been done since the discovery of Stuxnet to try to secure critical infrastructure in the US, many of the issues that made these systems vulnerable to attack in 2010 make them still vulnerable today.

In 2009 here in Washington DC, a metro train plowed into the back of another metro train that was stopped at a station during the afternoon rush hour. Sensors on the track should have indicated to any incoming train that another train was stopped at the station. Sensors on the front of the incoming train should also have detected the presence of the train at the station and alerted the driver or automatically slowed the incoming train. But the sensors failed to work and the driver noticed the stopped train too late and had trouble stopping the train manually. The collision killed nine people and injured 80 others. This incident, as far as anyone knows, wasn't the result of a cyberattack.

But this month CISA issued a critical security alert about a decade-old high-severity flaw in the braking system used by trains that hackers could exploit to cause a train to abruptly stop or derail. An attack like this could potentially result in the kind of outcome that occurred in 2009 of worse. The flaw exists in the protocol that devices located in the head and end of trains use to communicate with each other over radio to, among other things, engage the brakes and stop the train. The protocol employs a weak authentication, however, which means an attacker can also communicate with one of these devices as if they were a legitimate train device. They could send brake commands directly to a device, causing a train to halt or the brakes to fail.

The Association of American Railroads said it's developing more secure protocols and systems to replace the current devices and communication protocols. But the flaw was discovered by researcher Neil Smith back in 2012 and the AAR has ignored it since then, saying it was theoretical and without a real-world example to prove the flaw could be exploited in this way, it left the flawed system in place. Neil notified ICS-CERT years ago about the problem and together they tried unsuccessfully to convince the AAR to address it. But it was only after Smith and CISA recently threatened to go public with information about the flaw that the AAR announced it would be replacing the bad protocol. This won't happen, however, before 2027 at the earliest. The flaw can't be exploited over the internet — an attacker would need proximity to a train to communicate with it over radio frequency. But here's how Smith recently described it: "You could remotely take control over a Train's brake controller from a very long distance away, using hardware that costs sub $500. You could induce brake failure leading to derailments or you could shutdown the entire national railway system."

In my testimony today I'll focus first on explaining how Stuxnet operated so you can understand the level of expertise and sophistication that went into its unique design. Then I'll talk about the implications — how some of the tactics Stuxnet employed have been used by other threat actors since 2010, but also how the full capabilities demonstrated and hinted at by Stuxnet have not been realized yet in subsequent attacks. What I mean is that Stuxnet opened the door to a vast array of possibilities when it comes to attacking critical infrastructure, but threat actors

have so far refrained from deploying the most impactful and dangerous of these, though they no doubt possess the capability to use them.

The World's First Digital Weapon

Stuxnet was a first-of-its-kind attack in that it was the first *known* example of malicious code designed to leap from the digital realm to the physical realm to cause physical impact not on the computers it infected, but on the equipment and processes controlled by those computers. Unlike other malicious programs in the past that undermined the computer systems they infected, Stuxnet was targeting the industrial equipment those computers controlled — centrifuges — in order to have a kinetic impact on them and sabotage the enrichment of Iran's uranium. The same tactic and techniques can be used in other critical infrastructure environments to temporarily disrupt services that the public, government and military rely on daily; to permanently damage equipment; and, in some cases, to even cause loss of life — either directly by creating conditions that, for example, cause trains to collide, or indirectly by preventing patients from being treated at a hospital that doesn't have electricity.

Stuxnet was discovered the same year Operation Aurora was uncovered. Aurora was an espionage campaign, attributed to China, conducted against Google and dozens of other targets for intelligence-gathering purposes. Until Stuxnet was discovered, the only attacks we'd seen in the wild were either cases of cybercrime or espionage. When Stuxnet was first discovered, researchers believed it, too, was an espionage operation. This is because embedded in Stuxnet's code were instructions for it to search for the presence of Siemens Step 7 control software any time it infected a new system. The Siemens software is used to control and monitor all kinds of manufacturing and industrial processes, so researchers believed the attack was likely coming from China and was aimed at stealing the blueprints or configuration data for industrial plants so that China could emulate their designs. After reverse-engineering the code, however, researchers discovered that it was actually designed for sabotage.

This is significant, because attacks against critical infrastructure can be almost indistinguishable from espionage operations in their initial stages of infection. Both kinds of operations can use the same *types* of tools, or even *identical* tools, to gain initial access to a system, conduct reconnaissance to study the system or

network, and move laterally within the network to find the systems that contain the data an attacker seeks or that control the processes they want to affect. What's more, intrusions done initially for intelligence-collection purposes can morph into a disruptive or destructive operation simply by introducing malicious code or commands aimed at that purpose — meaning that an attacker may initially intend only to steal data from a system but then change course to damage or disrupt it as well, or to hand off access to the system to another actor who has the intention to disrupt or destroy. It can be difficult to discern the end goal of an intrusion until it's too late to stop it. I say this because a lot has been written recently about the Salt Typhoon and Volt Typhoon ongoing breaches of telecoms and critical infrastructure and attributed to China. These compromises don't appear now to be aimed at disruption or damage but could morph into such operations if China were to decide to use their presence in these systems for that purpose.

Returning now to Stuxnet and the Siemens software it sought, if Stuxnet found the presence of the Siemens Step 7 software on a system it infected, as well as evidence that the system was connected to a Siemens programmable logic controller — PLCs are essentially standalone computing devices that are used to control and monitor industrial equipment and processes — Stuxnet would then deposit its destructive payload on the PLC. But it did this only if it found a specific model and number of Siemens PLCs connected to the infected system as well as a specific model and number of other equipment Stuxnet was targeting. This was the precision part of Stuxnet that was aimed at ensuring that Stuxnet would not unleash its payload on any system except the intended target.

The Payload

Two known versions of Stuxnet were unleashed at separate times. The payloads in both of them operated similarly, though they impacted different parts of the centrifuges at Natanz. The first version of Stuxnet targeted the valves on the centrifuges, and the second version targeted the speed at which the centrifuges would spin.

With the first version of Stuxnet, once its payload was deposited on a Siemens PLC, Stuxnet would first sit on the device silently for 30 days and record the normal operation of the centrifuges as the PLC collected that data and sent it to

engineers at monitoring stations. The PLCs collected data about the temperature of the centrifuges the speed at which they were spinning; the pressure inside the centrifuges; and the state of the valves that managed the flow of gas into and out of the centrifuges, noting if they were open or closed.

At the end of the 30 days, the sabotage began. Stuxnet began to close the exit valves on some of the centrifuges to prevent gas from exiting the devices. Gas would continue to pour into the centrifuges, but could not get out. In some cases the valves it closed had already been chosen by the attackers and were hardcoded into Stuxnet. But Stuxnet also randomly chose some valves on the fly to avoid consistency. Natanz engineers might notice some of the valves malfunctioning and closing, but not be able to isolate the cause or see a pattern.

Stuxnet would close the valves for a period of two hours or until the pressure inside the affected centrifuges rose five times what was normal. During this time the valves were closed, Stuxnet took the data that it had recorded during the first 30 days, and fed it to monitoring stations so that engineers would not see what was occurring. To the engineers, the valves would have appeared to be open, and the pressure inside the centrifuges would have appeared to be normal. During this time, Stuxnet also disabled the safety system on the cascade — a cascade is a configuration of multiple centrifuges connected by a series of pipes. Safety systems on industrial control systems are designed to detect when a system or process is entering into an unsafe or abnormal condition. When the safety system senses this is occurring, it initiates an automatic shutdown of the affected components to alert operators and control the problem. Because Stuxnet disabled this system during its sabotage, however, the affected centrifuges did not shut down. At the end of the two-hour sabotage period, the centrifuges returned to their normal operation for another 30 days, when the same sabotage sequence would occur again.

There are two potential impacts from closing exit valves. By increasing the pressure of the gas inside the spinning centrifuges, the uranium gas would have begun to solidify and either slow down the spinning rotors or cause them to malfunction, potentially damaging the centrifuges and spoiling the gas.

The second version of Stuxnet operated in a similar manner. But this version was designed to alter the speed at which the centrifuges were spinning. When this version infected a PLC, it would sit on the device for 26 days recording the normal operation of the centrifuges and store that information. Then when the sabotage began, Stuxnet would increase the frequency controlling the centrifuges from 1,064 Hz to 1,400 Hz for fifteen minutes, then restore the centrifuges to the normal frequency. Stuxnet would then wait 13 days and cause the centrifuges to slow to 2 Hz for 50 minutes then restore the original frequency. During the sabotage, Stuxnet fed the recorded data to the monitoring stations so engineers would not see the change in frequency.

By increasing the frequency to 1,400 Hz, the attackers were pushing the centrifuges to the highest frequency they could withstand. The centrifuges Iran used were first-generation devices that had material defects, and the increased frequency would have caused them to deteriorate over time or spin out of control. By also slowing down the centrifuges to 2 Hz for 50 minutes, the attackers would have undermined the enrichment process itself. For enrichment, centrifuges have to spin at a high and uniform speed for uninterrupted lengths of time to separate the isotopes needed for nuclear fission from the rest of the material in the gas. By slowing down the centrifuges, any separated isotopes would have come back together with other particles in the gas, effectively undoing the enrichment. At the end of each enrichment cycle, Iran would have had less enriched gas than it expected to produce, and that gas would have been enriched to a lower level than Iran expected.

The engineers understood they were having problems with the centrifuges, but couldn’t determine the cause. This is because Stuxnet thwarted attempts to investigate. If the engineers tried to examine the code blocks on the PLCs to see if they had been corrupted in some way, Stuxnet intercepted the code blocks before they were displayed on the engineering station and scrubbed any malicious code from them so the engineers would see no change to them. If the engineers decided to wipe the existing code blocks from the PLC and load new ones, Stuxnet intercepted the fresh code blocks and injected its malicious code into them as well. In this way, Stuxnet remained undetected for three years.

The cyclical pattern to the sabotage, and the fact that only some centrifuges were impacted during each round of sabotage, tells us that the attackers were not looking to cause one-time catastrophic damage to the centrifuges and the enrichment process — this would clearly have been suspicious — but instead intended to cause only incremental impact over time that could not be easily detected. The aim was to slow the enrichment process in order to buy time for diplomacy to work and get Iran to the negotiating table over its nuclear program.

Stuxnet is believed to have first infected systems at Natanz in late 2007, and it remained undetected until 2010 when the attackers got reckless and added too many spreading capabilities to the second version of Stuxnet. These caused it to proliferate wildly out of control — which led to its discovery. But, again, because Stuxnet was a precision weapon, it didn't cause damage to other systems it infected.

I've provided all of these details about Stuxnet to demonstrate the high level of sophistication and expertise that went into this operation. Stuxnet required the attackers to have knowledge not only of the Siemens software and computer systems controlling the centrifuges, but also knowledge about the material and parts that formed the centrifuges and about the uranium gas and enrichment process in order to understand how their manipulation of the centrifuges would impact both. The attackers used model centrifuges and cascades made from the same material and design as the centrifuges in Iran, and built a makeshift cascade to test the impact the Stuxnet attack would have on the centrifuges and the enrichment process.

But as previously noted, other attacks on critical infrastructure would not need to have the same level of sophistication to cause considerable disruption or damage. The systems at Natanz were also air-gapped from the internet — meaning they were not directly connected to the internet. This made it difficult for the attackers to reach them. They needed an insider to physically and surreptitiously deliver the code for them. But many critical infrastructure systems are directly connected to the internet and have insufficient protections to prevent attackers from accessing them remotely.

In 2013 I wrote about a researcher who used an automated scanner to find systems connected to the internet that were using port 5900 (the port on a computer that is used for VNC and TeamViewer remote-management software). He found 30,000 connected systems that required no authentication to access them. This included two hydroelectric plants in New York, a generator at a Los Angeles foundry, a system for monitoring and controlling ventilation for underground miners in Romania, and the refrigeration system for a food service company in Pennsylvania that provided lunches to schools and other facilities. That was 2013. Surely, you'd think, this wouldn't still be the case years later. But in 2021, a water treatment facility in Oldsmar, Florida was hacked through its TeamViewer remote-management software over the internet. All of the computers at Oldsmar were connected to the internet without a firewall to protect them and limit who could access them, and all of them apparently shared the same password for the remote-management software.

Implications and Impact

One of the most significant impacts of Stuxnet was the awareness it brought to vulnerabilities in critical infrastructure that few had noticed before. The security community, largely focused before Stuxnet on IT networks — the systems used to run the business side of a company or industrial operation — had its eyes opened to a vast sector it had previously ignored: industrial control systems and the OT (operational technology) networks where they are deployed. Control systems consist not only of programmable logic controllers, but also SCADA systems and remote terminal units — devices that often sit in the field to operate and monitor equipment and processes that are distributed across large geographical distances, like electric substations. Stuxnet provided stark evidence that physical destruction of critical infrastructure – using nothing other than code – was not only possible but also likely. And once security researchers turned their sights on these systems, they found not only software security holes but also whole architecture problems that couldn't be fixed with a patch. With so many of the systems directly connected to the internet, cybersecurity suddenly became inextricably linked to national security.

The following is a small sample of the kinds of systems that PLCs and other industrial control systems operate. They control the opening and closing of cell doors and gates at high-security prisons; they manage the timing and

sequencing of traffic lights; they are used to manage HVAC systems in schools, hospitals and office buildings; they raise and lower bridges on waterways; they help route commuter and freight trains and prevent crashes; they control the temperature of food pasteurization processes to make food safe; they are used to control the temperature of furnaces in the manufacturing of steel and fiberglass; they control the flow and distribution of gas through pipelines; they control the operation of dams and water and sewage treatment plants; they operate and monitor the processes in chemical and pharmaceutical plants; and they help manage and control the distribution of electricity across the nation's grids — the critical infrastructure that undergirds all other critical infrastructure.

Years ago, industrial control systems were manually operated and were not connected to the internet, keeping them safe from remote attacks. But for efficiency purposes, these systems were digitalized. And then for varying reasons, ranging from regulatory requirements to ease-of-use, many of them were connected to the internet — without proper attention to securing them. Additionally, systems that once were highly complex and used proprietary software and protocols that were hard for attackers to access and study, have been simplified and standardized, making it easier for hackers to design attacks that can have widespread impact at scale. This is not news.

In 1997, after Timothy McVeigh blew up a federal building in Oklahoma, the Marsh Commission launched an investigation into the vulnerability of critical infrastructure to both physical and digital attacks. In their report, the commissioners warned against connecting critical systems for oil, gas, and electricity to the internet. "The capability to do harm … is growing at an alarming rate; and we have little defense against it," they wrote. Commands sent to the control computer at a power plant "could be just as devastating as a backpack full of explosives," they wrote at the time. "We should attend to our critical foundations before we are confronted with a crisis, not after. Waiting for disaster would prove as expensive as it would be irresponsible."

A second report also released in 1997 by the White House National Security Telecommunications Advisory Committee warned that the nation's power grid and utilities were vulnerable to digital attack. "An electronic intruder … could dial into an unprotected port and reset the breaker to a higher level of tolerance than

the device being protected by the breaker can withstand," investigators wrote. "By doing this, it would be possible to physically destroy a given piece of equipment within a substation."

But instead of heeding the warnings, critical infrastructure became more connected and more insecure.

After Stuxnet was discovered, experts expected to see a lot of copycat attacks against critical infrastructure. This surprisingly didn't occur. It wasn't until 2015 and 2016 that we saw the first Stuxnet-level attacks against critical infrastructure. These targeted Ukraine's electric grid to cause blackouts for a few hours at the height of winter. The attackers were able to take 60 substations offline in 2015, leaving about a quarter of a million customers without electricity. The attack was limited in scope — presumably it was simply done to send a message to Ukraine about who was in control of its grid not cause permanent disruption — but could have been much broader if the attackers had intended this. The subsequent attack next year showed the potential for this. The malware used in that attack, known as Industroyer and Crash Override, caused only a brief outage in parts of Kyiv. But the code was more advanced than the code used in 2015 because it had the potential to be automated so that once on a system, it could execute commands on its own such as opening circuit breakers, overwriting software or adapting to whatever environment it found itself on, without the need for direct control by the attackers. Whereas the 2015 outage required the attackers to be at the keyboards issuing a series of commands in real-time, the 2016 version could have unfolded automatically once the attackers unleashed the code.

Then in 2017, we saw an attack that went beyond disruption and destruction to target the safety system on critical infrastructure, as Stuxnet had done at Natanz. The so-called Triton attack was designed to disable the safety system at a petrochemical plant in Saudi Arabia. Presumably, the attackers intended to use it in conjunction with an attack that would have caused a chemical spill or some other dangerous condition at the plant and they wanted to prevent the equipment from automatically shutting down to contain the danger. But fortunately there was no accompanying attack in this case, and the code targeting the safety system contained a flaw that caused the safety system to trigger automatic shutdowns of the plant, alerting engineers to its presence. It's an attack that could have had a

potentially deadly impact if the attackers had intended this and if they had not made a mistake.

Triton wasn't a fully developed and tested attack tool yet. But the expansive Pipedream attack platform discovered in 2022 was. Researchers at the security firm Dragos say it had the potential to cause disruption or destruction and appeared to be focused on electric and oil and gas facilities — liquified natural gas systems in particular. It could be modified, however, for use against any industrial environment and had the ability to disable or brick control systems or undermine safety systems in ways that could potentially endanger lives if an attacker can cause chemicals to spill or cause equipment to catch fire or explode. This impact can be multiplied if disabled safety systems prevent engineers from being alerted to a dangerous condition when it first starts to unfold or prevent the systems from going into automatic shutdown to contain the damage and impact.

Since 2017, hackers have increasingly been targeting critical infrastructure and industrial control systems — whether cybercriminals infecting them with ransomware to extort the infected organizations, nation-state actors targeting them to cause disruption or hacktivists impacting them to send a message. In 2022, the state-owned Khuzestan Steel Company in Iran had to halt operations after being hit with a cyberattack. The company claimed it thwarted the attack and no damage or disruption occurred. But a hacktivist group believed to be tied to Israel claimed credit for the attack and published CCTV footage as proof that it did have an impact. The video, purportedly taken from inside the plant, showed a fire breaking out from malfunctioning equipment that spilled molten steel, evidently a result of the cyberattack. Regardless of whether the hacktivist claim is true, it is possible that such an attack could result in spillage and a fire.

Small critical infrastructure organizations are more vulnerable to attack due to the fact that they tend to have insufficient funding to hire security staff and replace outdated insecure systems. By contrast, large well-resourced facilities tend to have redundant systems that make them more resilient to attack so they can prevent disruption and downtime or limit their impact. But this is not always the case. The ransomware attack against Colonial Pipeline in 2021 revealed that this company did not have a CISO in place at the time of the attack, had seemingly failed to properly segment its IT and OT networks (requiring the company to shut

down the pipeline to prevent the malicious code from spreading to its OT systems) and prior to the attack ignored warnings about lax security as well as government alerts about attackers targeting pipelines.

The ransomware struck around 5am on May 7, and by 6am the company had shut down its 5,500-mile pipeline. By late afternoon CEO Joseph Blount had decided to pay the ransom, which was sent to the hackers the next day. He later said they shut down the pipeline out of fear that the ransomware might spread from the IT to the OT network, taking control of the pipeline out of their hands. The pipeline was down for nearly a week and resulted in a cascade of effects the company had no direct control over — panic buys and hoarding triggered by consumer reaction to the outage. The hack didn't inflate prices and create a fuel shortage, but consumers responding to it did.

When Colonial Pipeline was hit, many were surprised at how quickly the company paid the $4.4 million ransom. Surely a business as big and critical to the US economy — Colonial Pipeline supplies 45 percent of fuel to the East Coast, which amounts to about 2.5 million barrels daily — had sufficient backups and a response plan in place to recover from the attack without needing to pay the ransom. The company did have an emergency-response plan, the CEO told lawmakers on Capitol Hill after the attack, but it didn't include a game plan for ransomware — even though ransomware actors had been targeting critical infrastructure since 2015.

Colonial Pipeline was caught off guard. But the warnings were there if the company had been paying attention.

There had been some 400 ransomware attacks against critical infrastructure the previous year; and between November 2013 and June 2022, there were nearly 1,300. These included attacks on oil and gas facilities. The ransomware operators weren't just targeting IT systems in critical infrastructure — they were going after OT systems to disrupt critical processes.

In 2020, the year before Colonial Pipeline was hit, the security firm Mandiant reported that seven different ransomware families had struck industrial organizations since 2017, resulting in significant disruptions and delays in

production as well as the delivery of goods and services. Ransomware actors were also becoming increasingly sophisticated, Mandiant reported, conducting internal reconnaissance of their victims to determine which systems were the most vital to production, in order to increase the odds that a victim would pay. The ransomware operators actually put together a “kill list” of more than 1,000 processes that ransomware operators could choose to halt to increase the odds of being paid.

If this wasn’t enough warning, that same year, DHS’s Cybersecurity Infrastructure and Security Agency published an alert warning specifically about ransomware attacks targeting pipelines. It described an attack against a natural-gas compression facility that began with a phishing campaign that infected the IT network, then spread to the facility’s improperly segmented OT network, preventing staff from obtaining real-time data from control and communication systems and forcing the company to shut down operations for two days. The plant didn’t have a response-plan for cyber attacks in place, and in its alert, CISA advised pipeline and other critical infrastructure owners to create a response plan, conduct red team exercises to simulate attacks and test internal responses, put backups offline or on fully segregated networks to keep them from being encrypted along with the rest of their systems, and build redundant workflows to maintain critical operations in the event of an attack. A year later, ransomware struck Colonial Pipeline.

The attackers got in through an employee password for the company VPN that the employee had apparently re-used for other systems. Mandiant later discovered it in a batch of passwords leaked online from a different data breach, though it’s not clear if the Colonial Pipeline hackers obtained it this way. The VPN account was a legacy system the company no longer used but had failed to disable. And because Colonial Pipeline didn’t have multi-factor authentication enabled on the account, the attackers were able to get in using just the employee’s username and password.

The company told the Associated Press that its IT and OT networks were segmented, but if Blount made the decision to shut down the pipeline because the company was afraid the ransomware would spread to the OT network, this suggests the company wasn’t as confident in the segmentation as he indicated.

He also said his company had the ability to operate the pipeline manually, but only, unfortunately, on a small scale if a portion of the pipeline went down — not in a scenario in which the entire 5,500 miles of pipeline were shut off.

In 2018, three years before the ransomware attack, an audit of Colonial Pipeline systems found that it was deficient in security best practices. Robert Smallwood, whose consulting company conducted the audit, called Colonial Pipeline's information management practices "atrocious" and said the company had a patchwork of poorly connected and secured systems and lacked security awareness.

In 2022, CISA released a lengthy list of basic security guidelines for pipelines: use strong perimeter controls to isolate ICS/SCADA systems and networks from corporate networks and the internet; limit communication leaving/entering these perimeters; use multi-factor authentication; have a cyber incident response plan in place; and maintain good offline backups.

When these came out, many wondered why CISA would distribute a list full of basic guidelines — especially after years of red flags about threats to critical infrastructure. But Colonial Pipeline — which, remember, had no CISO at the time of the hack — showed that companies were still not doing some of the basics to secure their systems and ensure they would be resilient in an attack.

Several years ago, CISA launched a "More Than a Password" campaign to increase adoption of multi-factor authentication and called the absence of MFA "exceptionally risky," particularly for critical infrastructure. A study by Google and two universities found that MFA can block up to 99% of bulk phishing attacks and about 66% of targeted attacks. Yet a survey published by Trellix found that 75 percent of respondents in the US oil and gas sector had not fully deployed MFA. Over half of them blamed a lack of in-house cyber skills for failing to implement it.

So although there has been a lot of focus from the government in establishing new security guidelines and mandates and reporting requirements for railways and pipelines and other critical infrastructure, it's not clear how these industries will reach basic levels of security without budgets and skills — and even with those, it's not clear how long it will take to get them up to speed. The fact that

there aren’t more attacks against critical infrastructure isn’t because the systems are secure.

Testimony like this often ends with some sort of call to action. I don’t have any specific prescriptions to suggest because I believe my fellow panelists will do that. My goal here has been to bring attention to some issues around critical infrastructure that have been simmering for two decades but are far from being resolved, even though we’ve had decades to address them and events like Stuxnet, the Ukraine power grid hack and the Triton assault against the petrochemical plant in Saudi Arabia to illustrate the direction the US is headed if the problems aren’t addressed.

Thank you again for this opportunity to speak with you about this issue.